**Pseudo code for outlining Personal Finance Tracker**

BEGIN

#Global list

Initialize transactions to []

FUNCTION add\_transactions ()

DISPLAY “Enter Transaction Details”

GET amount from user as input.

GET category from user as input.

GET type (“Income or Expense”) from user as input.

GET date(“DD-MM-YYYY”) from user as input.

Set transaction to [amount, category, type, date]

Add transaction to transactions

DISPLAY “Transactions Added Successfully”

ENDFUNCTION

FUNCTION view\_transactions ()

IF transactions list is empty THEN

PRINT “No transactions to view”

ELSE

DISPLAY “Transactions are listed below”

FOR view\_ transaction in transactions DO

DISPLAY “Amount:” , view\_ transaction [0]

DISPLAY “Category:” , view\_ transaction [1]

DISPLAY “Type:” , view\_ transaction [2]

DISPLAY “Date:” , view\_ transaction [3]

NEXT transaction

ENDIF

ENDFUNCTON

FUNCTION update\_transaction ()

IF transactions list is empty THEN

PRINT “No transactions were found to update”

ELSE

PRINT “Enter transactions index to update”

GET index from user input

IF index is valid THEN

PRINT “Enter new transactions details”

GET correct amount from user input

GET correct category from user input

GET correct type (“Income or Expense”) from user input

GET correct date (“DD-MM-YYYY”) from user input

Set transactions[index] to [correct amount, correct category, correct type, correct date]

save\_transactions ()

DISPLAY “Transaction Updated Successfully”

ELSE

PRINT “Invalid index to process”

ENDIF

ENDIF

ENDFUNCTION

FUNCTION delete\_transaction ()

IF transactions list is empty THEN

DISPLAY “Transactions list is empty to be deleted”

ELSE

Prompt for index of transactions to delete

GET index from user input

IF index is valid THEN

Delete transactions[index]

save\_transactions ()

PRINT “Transaction deleted successfully”

ELSE

PRINT “Invalid index to process”

ENDIF

ENDIF

ENDFUNCTION

FUNCTION display\_summary

Set total\_income to 0

Set total\_expense to 0

FOR transaction in transactions DO

IF transaction[2] == “ Income” THEN

total\_income = total\_income + transaction[0]

ELSE

total\_expense = total\_expense + transaction[0]

ENDIF

NEXT transaction

IF total\_ income > total\_expense THEN

Compute Net\_income = total\_income - total\_expense

ELSE

Compute Net\_expense = total\_ expense - total\_ income

ENDIF

PRINT total\_income

PRINT total\_ expense

IF total\_ income > total\_expense THEN

PRINT Net\_income

ELSE

PRINT Net\_expense

ENDIF

ENDFUNCTION

FUNCTION save\_transactions ()

OPENFILE “transactions.json” for WRITE as F1

Serialize the transactions list to JSON

WRITE the serialized JSON data to F1

CLOSEFILE F1

DISPLAY “Successfully saved”

ENDFUNCTION

FUNCTION load\_transactions ()

OPENFILE “transactions.json” for READ as F1

Deserialize the transactions list to python

CLOSEFILE F1

DISPLAY “Successful”

ENDFUNCTION

FUNCTION main\_menu ()

load\_transactions ()

WHILE True DO

PRINT “Personal Finance Tracker”

PRINT “1. Add Transaction”

PRINT “2. View Transaction”

PRINT “3. Update Transaction”

PRINT “4. Delete Transaction”

PRINT “5. Display Summary”

PRINT “6. Exit”

Prompt for choice as input

GET choice from user input

IF choice == 1 THEN

add\_transactions ()

ELSE IF choice == 2 THEN

view\_transactions ()

ELSE IF choice == 3 THEN

update\_transactions ()

ELSE IF choice == 4 THEN

delete\_transactions ()

ELSE IF choice == 5 THEN

display\_summary ()

ELSE IF choice == 6 THEN

PRINT “Exiting the program and saving….”

ELSE

PRINT “Invalid choice. Please try again”

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

ENDIF

ENDWHILE

ENDFUNCTION

main\_menu ()

END